JWT Protection - Scalia CDN

As the use of Content Delivery Networks (CDN) continues to grow, the need for security measures to protect assets on the network has become increasingly important. One way to secure access to these assets is through the use of JSON Web Tokens (JWTs).

JWT tokens are a type of token that provide a secure and easy way to transmit information between parties. They are used to authenticate and authorise access to resources in a distributed environment.

For our CDN, we have developed a feature that allows customers to use JWT tokens to secure access to assets on the CDN. To use this feature, customers need to enable it in the Scalia Portal and supply the JWT secret for the HS256 encryption. Future releases will include configurable encryption algorithms and the use of public keys. After enabling, each CDN URL has to be followed by the `GET` parameter `token`, followed by the generated JWT token.

Enabling JWT Protection

To use this feature, please follow a few simple steps:

1. Enable the feature in the Scalia Portal. This can be done by logging in to your account and navigating to the configuration section of the preferred CDN entry.
2. From the configuration wizard toggle “JWT Protection” and supply a secret key. This key should be kept secure and not shared with unauthorised individuals.
3. After selecting “Save“, the CDN will not serve any assets without having proper authentication in the form of a token.

Generating a Token

In most cases the client application will generate and supply a end user with an JWT Token. Scalia JWT Protection requires a token with the following format.

Header	Required	Value	Availability
alg	Yes	HS256	Since v0.1.0
typ	Yes	JWT	Since v0.1.0

Claim	Required	Remark	Availability
`iat`	Yes	Unix timestamp; time of generating the Token	Since v0.1.0
`nbf`	Yes	Unix timestamp; time when token will be valid	Since v0.1.0
`exp`	Yes	Unix timestamp; time when token will expire	Since v0.1.0
`file`	Optional	String; path of the asset which will be bound to the token	Since v0.3.0
`ip`	Optional	String; IP which will be bound to the token	Since v0.3.0
`geo`	Optional	String; ISO format country geofence which will be bound to the token	Coming Soon

We recommend using the available tooling at JWT.io, you may find an example of an JWT token with all the above Claims here.

Evaluation

Tokens are evaluated in sequence, with each claim being evaluated one at a time. If any of the evaluations appear invalid, the access to the asset will be rejected. The following is a more detailed explanation of the token evaluation process:

1. Signature Verification: The first step is to verify the token signature using the secret key. This ensures that the token has not been tampered with and was issued by a trusted party.
2. Not Before Check: The not before (`nbf`) claim is checked to ensure that the token is not being used before its intended time. If the current time is before the `nbf` claim, access to the asset will be denied.
3. Expiration Check: Next, the expiration time (`exp`) claim is checked to ensure that the token has not expired. If the token has expired, access to the asset will be denied.
4. File Check: The `file` claim is checked to ensure that the token was issued for the requested asset. If the file does not match the request path, access to the asset will be denied.
5. IP Check: The IP (`ip`) claim is checked to ensure that the token was intended for the current requester. If the remote IP of the requester does not match `ip` , access to the asset will be denied.
6. Additional Custom Claims: Will follow soon in new releases of the Scalia CDN

Implementation

After generating the token and signing it with the secret key it can be used within the Scalia CDN. To use the feature, customers must append the token parameter to the CDN URL followed by their generated JWT token. For example:

```
https://cdn.example.com/assets/image.jpg?
token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9....
```

By following these simple steps, customers can quickly and easily enable the JWT token authentication feature and enjoy secure access to their assets on the CDN. This feature provides an additional layer of security to ensure that only authorised individuals can access the files on the network.

Overall, the use of JWT tokens is a secure and easy way to protect assets on a CDN. With the JWT token authentication feature, customers can enjoy peace of mind, knowing that their assets are secure and protected from unauthorised access.